Improving Geolocation of Social Media Posts

Elizabeth Williams[1], Jeff Gray[2], Brandon Dixon[3]

University of Alabama

Box 870290 Tuscaloosa, AL 35487

P: 205-348-6363

F: 205-348-6959

[1]eawilliams2@crimson.ua.edu, [2]gray@cs.ua.edu, [3]dixon@cs.ua.edu

Abstract

Pervasive social systems often take advantage of geographical information to provide real-time information to users based on their location. However, due to privacy concerns, many social media users do not share their exact geographical coordinates. In this paper, we describe our technique that predicts locations of posts that are not associated with explicit coordinates, a process called geolocation. Existing research has utilized the content of a post as well as the post author's social media relationships with other users to estimate location. Our research provides a novel approach to geolocation by combining multiple techniques, as well as adding a new technique: estimating location by clustering similar social media posts that are centered in a geographical area.

Keywords

social media, geolocation, pervasive computing

1. Introduction

Pervasive computing systems are often adaptive and react to changes in a user's environment. Social media can provide a deep insight into a user's environment by explicitly defining what a user is doing, what relationships a user has with other users, and, sometimes, where

a user is located. Knowing where social media users are located can provide a more adaptive user experience, as well as the ability to determine topics that are being discussed in different geographical regions [1]. If these topics are clustered in the same geographical area of a user, the topics are likely to be more important to that user. For example, a news or social network site can customize its content to a user's location.

However, on the social networking site Twitter, as few as 0.87% of tweets are geotagged, or associated explicitly with geographical coordinates [2]. In our previous research, we found that only 2.63% of tweets we analyzed were tagged with explicit location information [3]. Without many tweets being associated with a specific location, adaptive systems are unable to take advantage of the location of the tweets to provide location-specific information. This paper describes our system, GeoContext Locator (GCL), which provides a method for predicting the location of, or geolocating, social media posts.

Like most related research, we chose Twitter[1] to implement GCL for several reasons. First, Twitter allows users to write short posts with freeform text that can be analyzed to discover locations. Also, Twitter allows users to connect locations to posts in two ways. Users can attach geographical coordinates directly to the post, called geotagging. Alternatively, Twitter users can tag tweets using Twitter Places[2], which allows a user to tag a post with the name of a location. This tag also includes a bounding box of geographical coordinates around the location. In addition to location information, Twitter allows users to establish relationships with other users by following another user. Following someone means that the user will receive updates on their Twitter home page when the other person publishes a post. For a specific user, Twitter describes users who the specified user follows as *friends* and users who follow the specified user as *followers*.

[1] http://dev.twitter.com
[2] http://dev.twitter.com/overview/api/places

Although we chose to utilize Twitter for GCL, it can be adapted easily to any other social network. GCL simply processes JSON objects from a social media stream that have content and location information, so any other social network or information provider that attaches geographical coordinates to shared information could be used. In this paper, we describe our system, GCL, for geolocating a stream of tweets. Our system is unique in several ways:

1. GCL combines analysis of the content of the tweet, the location specified on the user's account, and locations of the user's friends and followers to perform geolocation. Previous research has not utilized all of these aspects together to discover a tweet's location.

2. GCL also includes a novel approach to geolocation by estimating a tweet's location by analyzing the locations of tweets with similar content in real-time.

3. GCL utilizes cognitive computing resources [4] AlchemyAPI[3] and Dbpedia[4] to extract locations from plain text.

The structure of this paper is as follows: first, we discuss existing work in the area of geolocation of social media posts. In Section 3, we define our approach, GCL, for predicting tweet location. We then describe our experimental setup and results in Section 4. Finally, we conclude with a discussion of future work.

2. Related Work

Existing research has been focused in mainly two areas: geolocation based on the content of the social media post and geolocation based on the relationships of the user with other users on the social media network. We first discuss research based on the content of the post.

[3] http://www.alchemyapi.com/
[4] http://dbpedia.org

2.1 Content-based geolocation

Several approaches are based on comparing a tweet to previous tweets with known locations to discover similarities between the tweets. Tweets that are determined to be similar can be inferred to have similar locations. A significant number of geotagged tweets occur as a result of the user having other location-based social networks, such as Foursquare, that send automatic geotagged messages to their Twitter account. Watanabe et al. [5] created a database from tweets that were posted via Foursquare. They were then able to use the database to look up place names in non-geotagged tweets and predict the location of the non-geotagged tweets. Ikawa et al.'s [6] approach for predicting user locations involves extracting keywords from tweets in a training set. Keywords are then extracted from the test set tweets, and the keywords are compared to those in the training set. Cosine similarity is computed between the keywords, and the location associated with the keyword set in the training set is estimated as the location of the tweet in the test set.

Several approaches predict locations within a grid cell rather than as geographical coordinates. Wing and Baldridge [7] ran their geolocation algorithm on Wikipedia documents, rather than a more traditional type of social media platform such as Twitter. However, like the previously described related work, the authors also utilize the content of the document to predict a location of the text. Their approach divides the Earth into varying sized cells and predicts a cell for each document. Their model calculates the distribution of words over different locations and compares the word distribution of each document to the word distribution of each geographic cell, eventually choosing the cell with the highest similarity. Baldwin et al. [8] also predict the location of the author of each post within a grid cell on a map. Their approach utilized a naive Bayes classifier to approach the problem of geolocation. They split each Twitter post into tokens and consider each token as a feature in the classifier.

Some approaches utilized existing web services or other APIs in order to perform geolocation. Jaiswal et al. [2] utilized a named-entity extraction module, ANNIE, to extract

possible locations from the content of Twitter posts. The locations were then mapped to geographical coordinates (a process called geocoding) using the geonames.org webservice. In this approach, the authors take into account temporal information present in the tweet content. For example, if the word "tomorrow" is present in the tweet, the location mentioned in the tweet will be predicted to occur one day after the timestamp of the tweet. Baucom et al. [9] discussed their approach for analyzing how Twitter models the real world through the example of social media discussions about a basketball game. In order to perform the analysis, the authors geolocate tweets by passing the location associated with each user's account (not the location associated with each tweet) through the Google Maps geocoordinates API.

Several approaches involved creating models to calculate the distribution of text over geographical areas. Han et al. [10] experimented with several algorithms for location prediction, including a generative Naive Bayes model and KL divergence [11]. The authors attempt to predict the "home location" of the user associated with each tweet and assume that the users remain in the same location throughout the dataset. Hong et al. [12] outlined their approach for defining a model that describes the global distribution of topics. The geographical location portion of their model is a collapsed Gibbs sampler [13] for locations. Yuan et al. [14] described their model, EW^4 that uses a generative process to model tweets along with their day, time, words, and location. The model is able to predict user location by incorporating the temporal aspect of the tweet. It utilizes both location identifiers (e.g., text descriptions of the location) and geographic coordinates in the model to better predict location. Cheng et al. [15,4] also modeled the distribution of words over locations to discover "local words," or words that are used more frequently in a localized region. They extracted words that are used frequently in one point and whose usage drops off rapidly around that central point. Tweets containing "local words" are predicted to be in the locations where "local words" occur.

We next discuss the research based on the relationship graph of the user within the social network.

2.2 Relationship-based geolocation

Backstrom et al. [1] described their approach for predicting location based on the relationships of the user on a social network. When analyzing the relationship graph of a social network user, the mean or median location of the user's friends may not be accurate. For example, if the user has one friend living far away, the user's location will be inaccurately influenced by that "outlier" friend. The authors constructed a probabilistic model that determines the likelihood of a given location being the actual location of the user. The model is based on the probability that each of the user's friends would have a friend living in that location. To compute the location prediction, the model computes the likelihood that each friend's location is the user's location.

McGee et al. [16] extends Backstrom et al.'s approach of using relationships to predict location by including tie strength, or a measure of how much two users interact, in their prediction. Unlike previous work on relationship-based geolocation, this approach does not treat friends equally. The authors construct a model consisting of a tree classifier and a maximum likelihood estimator to predict location.

Like Han et al. described in the content-based geolocation section previously, Li et al. [17] were interested in predicting users' home locations. However, they utilize the relationship graph of the user rather than the content of the tweet. Their model can analyze the likelihood that the user is in various locations of his friends based on the probability that an edge between the user and the friend exists without the user living in the friend's location. The model can also take into account the "influence scope" of the user. For example, a celebrity Twitter user is more likely to have followers in distant locations.

Similar to our approach, Rout et al. [18] utilized Dbpedia as a resource for looking up location information. They used regular expressions to extract locations from the user account locations of each user's friends. Like Han et al., the authors also take into account the probability of the existence of an edge between the user and a friend depending on the population size of the friend's city. Their model also considers whether an edge in the relationship graph is unidirectional or bidirectional (i.e., whether the friendship is reciprocated).

The existing approaches described in this section are the state of the art in geolocation research. Our pipeline for predicting location differs from these approaches in several ways. GCL combines both aspects of geolocation research, content-based geolocation and relationship-based geolocation, in order to gather location information from multiple sources within the social media post. Also, we introduce topic-based geolocation, which geolocates tweets with other tweets of the same topic.

3. GeoContext Locator

For our approach to geolocation, GCL predicts the current location of a tweet, which corresponds to the current location of the user at the timestamp of the tweet. This differs from several of the existing research approaches that predict the user's home location. GCL is useful for geolocation on a mobile device where the user's current location is often more important than the user's home location. Therefore, if a user tweets while on a trip or out of the region of their home location, GCL will predict the location the tweet was sent, not the location where the user normally resides.

We utilized the "Gardenhose" stream of tweets from Twitter, which provides a stream containing a small percentage of all public tweets. A tweet received from the stream is represented as a JSON object. The object consists of the actual content of the tweet and metadata about the tweet, such as geographical coordinates if the tweet is geotagged and a timestamp. The object also

contains metadata about the author of the tweet, such as the username, account location, account description, and more. Tweets that are "retweets" (tweets that are re-published from other users) contain extra metadata, but as we do not consider retweets different from non-retweets, we do not go into detail regarding retweet metadata. An important detail regarding tweet objects is that although few tweets are geotagged and contain location information attached to the tweet itself, many more users provide account locations, which are locations attached to the user account rather than a specific tweet. Account locations can be freeform text, so they do not follow any convention in terms of representing location, which makes geocoding these locations difficult. Moreover, because account locations can be any text, some users provide a location that is not a true location, such as "Cloud 9." Users also have the option of leaving the account location blank.

Unlike most previous work, GCL combines the content of the tweet, the user account location, relationship graph, and the extracted topic of the tweet to predict the tweet's location. Tweets collected from the stream are processed through the GCL pipeline, shown in Figure 1. In each step, GCL extracts location information from various aspects of the tweet object. GCL then attempts to geocode the raw location information, or convert it to geographical coordinates. Throughout the pipeline, GCL stores an object for each tweet consisting of all estimates of coordinates from each step. If the geocoding process is successful, the resulting coordinates are stored in the list of predictions, along with the step from which they resulted (i.e., "user location" or "content"). At the end of the pipeline, the list of predictions is analyzed, and the most likely set of coordinates is chosen to be the location prediction for that tweet.

In the following subsections, we describe each step of the GCL pipeline for predicting tweet location.

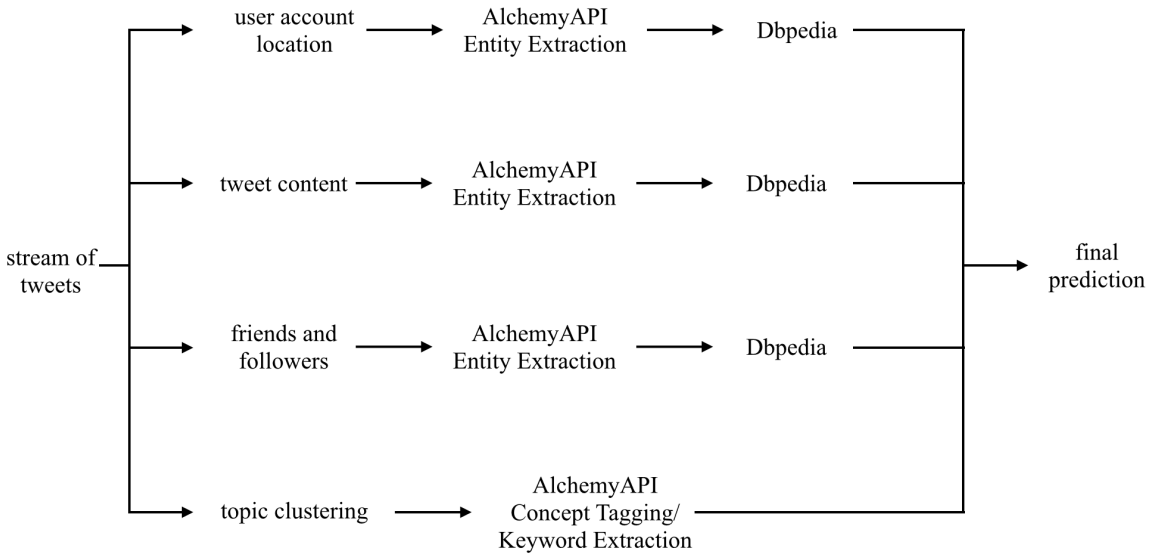

Figure 1

3.1 User Account Location

The first portion of the tweet that GCL checks for location information is the user account location. As described previously, the user account location is a location attached to a user's account, rather than to each individual tweet. On a user's Twitter account, the user can input any freeform text for the user location, or the user has the option to leave the location blank. An example of the user account location is shown in Figure 2. If the user location is blank, or null, GCL ignores this field and proceeds to the next step, because no location information can be extracted.

Because the user location is freeform text, GCL must be able to extract text that represents locations from the field. GCL performs this extraction in multiple steps. Although some of the steps can be redundant and result in the same location being extracted multiple times, none of the

methods used by GCL are perfect in extracting location information, so multiple techniques are needed in order to provide more coverage and ensure locations are discovered in the text.

Tweet Data	Example Data
user account location	New York, NY
tweet content	Stuck in a traffic jam on 5th Ave.
friends and followers	New York City, New York, London, NYC
topic	traffic

Figure 2

3.1.1 Extracting Location Entities from AlchemyAPI

First, we utilized AlchemyAPI's Entity Extraction API[5] to look for text that resembles locations. The Entity Extraction API takes in a piece of text as input and returns a ranked list of named entities, such as people, organizations, or locations. As each tweet comes in through the stream, we pass the user account location associated with the tweet to the Entity Extraction API and receive back results indicating whether the account location contains any named entities.

If no results are received from the Entity Extraction API, we move on to the next technique for discovering location data, described in Section 3.1.2. If results are received, this indicates that the Entity Extraction API has been able to extract some entity information. However, not all entities found by the API are related to location. For example, the API extracts numerical values

[5] http://www.alchemyapi.com/api/entity/textc.html#rtext

and Twitter mentions (i.e., a tweet that contains another user's username, also known as their Twitter handle). Neither of these entities are useful to GCL in determining locations at this time, so we only consider results for entities that contain location information. Each discovered entity is associated with a type, and the types considered by GCL are "City," "Region," "Facility," "StateOrCounty," "Organization," "Company," and "GeographicFeature." "City," "Region," "StateOrCounty," and "GeographicFeature" are clearly types of entities related to the user's location. "Facility" is often related to locations such as sports stadiums. "Organization" and "Company" can be entities such as sports teams, which are often location-specific.

For each entity extracted from the user location, GCL checks to see if the type of the entity matches any of these types GCL considers for location information. If there is a match, GCL then attempts to extract geographical coordinates from the entity results. Some results simply contain geographical coordinates contained within the result. In this case, the coordinates are extracted and added to the tweet's list of predicted locations. Some results do not contain geographical coordinates, but contain a link to a Dbpedia entry of the entity. In this case, GCL sends a POST request to the Dbpedia entry for the entity and extracts any coordinates found in the entry. We provide more detail about requesting and extracting coordinates from Dbpedia entries in Section 3.1.2. Lastly, some results do not contain any disambiguation information, but simply the entity text that was extracted from the user location and the type of the entity. In this case, GCL creates a URL to Dbpedia that is similar to the URLs received from results that contain links to Dbpedia entries by prepending "http://dbpedia.org/resource/" to the entity text. GCL then follows the same procedure as if the results contained a Dbpedia link by sending a request to the entry and extracting any coordinates the entry contains.

After any geographical coordinates are extracted, the coordinates are added to the list of predicted locations. Next, the process continues, attempting to extract location data from the user account location.

3.1.2 Extracting Location Information from Dbpedia

We determined through manual observation that the AlchemyAPI Entity Extraction API did not recognize 100% of locations in the stream of tweets. Therefore, we utilized Dbpedia as a technique for discovering additional locations in the user account location field. Dbpedia is a database consisting of structured information extracted from Wikipedia. More specifically, Dbpedia contains information found in the sidebar information boxes on Wikipedia. This makes it ideal for discovering entities such as cities or states mentioned in text.

GCL executes the user location through the Dbpedia extraction step in three passes. In the first pass, the entire user account location field is used. Any punctuation is removed except commas, as other punctuation affects the request. Spaces are also replaced with underscores. GCL then creates a Dbpedia URL by prepending "http://dbpedia.org/resource/" to the user account location text.

GCL sends a POST request to the URL and attempts to retrieve results from Dbpedia. If the request succeeds, this indicates that an entity exists within Dbpedia of the user location. Some Dbpedia entries are pointers to another entry; in this case, the entry is disambiguated to another. For example, the entry for "Tuscaloosa" disambiguates to "Tuscaloosa,_Alabama." If the results received from the request indicate that the entity disambiguates to another entry, GCL sends another request to the disambiguated entry's URL.

After results of an entry are received that do not disambiguate, GCL extracts any geographical coordinates that are available from the entry. Any entry that is a location, (e.g., city, state, or region) contains coordinates. The coordinates are added to the list of location predictions stored with the tweet.

In the second and third pass, GCL performs the same process of sending a request to Dbpedia and extracting geographical coordinates but with different text used as the entity. In the

second pass, GCL removes all punctuation and splits the user account location text by spaces and sends each token as the request to Dbpedia. We chose to split the text by spaces, as this allows GCL to discover locations from text such as "Alabama, y'all," in which extra words are present in the user location that would result in a bad request in the first pass. In the third pass, punctuation is removed, the text is split by spaces, and then two tokens are concatenated with a space in the middle. This ensures that user locations such as "My office, Beverly Hills" is analyzed properly. "Beverly Hills" can be extracted, while extra words such as "My office" are effectively ignored. We chose not to perform more passes where more than two tokens are concatenated at this time, as it did not seem to affect our results significantly because all fields in a tweet are kept relatively short.

After processing the user account location, GCL passes the tweet and the list of predicted location coordinates to the second step in the pipeline, which is tweet content analysis.

3.2 Tweet Content

The next step in the GCL pipeline extracts location data from the actual content of the tweet. Twitter limits the tweet text to 180 characters, so each tweet is relatively short. An example of tweet content is shown in Figure 2. Similar to the user account location, Twitter allows the tweet content to be freeform, and users are able to use any punctuation or character they choose.

GCL follows the same procedure when analyzing tweet content as the user account location. First, GCL passes the content of the tweet to AlchemyAPI's Entity Extraction API. As described in Section 3.1, the Entity Extraction API discovers entities present in the text. If the entity is related to a location, GCL is able to geocode the entity name into geographical coordinates, either directly from the Entity Extraction results or indirectly through Dbpedia.

After utilizing the Entity Extraction API, the tweet content is converted to a Dbpedia URL and the corresponding Dbpedia entry is requested. Less of the requests sent from the tweet content

succeed than the user account location, because, although not all users put a valid location in the user account location field, many do, while most words in the tweet content are not words related to location. As with the user account location, three passes are performed over the content. First, the entire tweet with punctuation removed is requested as a Dbpedia entity. In the second pass, GCL removes punctuation and tokenizes the tweet content by splitting it by spaces. Each token is requested as a Dbpedia entity. Lastly, GCL removed punctuation, splits the content text by space, and concatenates each two consecutive tokens together with a space in between. Each token combination is requested as a Dbpedia entity.

After each request, if the request is valid and the results contain geographical coordinates, the coordinates are added to the list of possible location predictions for the tweet. The tweet and list of predictions are then passed to the third step in the GCL pipeline.

3.3 Friends and Followers

The third step in the GCL pipeline for predicting tweet location analyzes the friends and followers of the tweet author. Within the Twitter social media platform, users can follow each other, which means they receive updates on the Twitter home screen when a followed user posts a tweet. Given a specified user, users that the specified user follows are called "friends." Users that follow the specified user are called "followers."

As mentioned in Section 2, some existing geolocation approaches utilized the relationship graph of a user to predict location. Like these approaches, we also analyze the relationship graph of each tweet author. McGee et al. [16] discovered peaks in the distribution of friends and followers around the area where the user lives. This result shows that although Twitter allows users from all over the world to communicate, users tend to have a collection of friends and followers near their same location. The friends and followers of a user are valuable for finding the user's home location, especially in the case where the user does not provide a user account location or the

account location does not contain a valid location. However, the friends and followers location is not always accurate, especially in the case where a user is on a trip or away from the home location. It is for this reason that we combine multiple techniques for predicting location.

We are able to collect the friends and followers of each user through the REST APIs provided by Twitter[6]. The REST APIs allow the lookup of friends and followers based on a Twitter ID. An object representing the friend or follower, respectively, is returned when requested via the API. The object contains metadata about the friend or follower, including username (also known as Twitter handle), description, and user account location. Following existing approaches, we utilize the user account location for each friend or follower when calculating a predicted location.

After collecting the user account locations for friends and followers of the tweet author, we follow a two-step process that is similar to the user location and tweet content steps in the GCL pipeline. GCL first passes each user location of each friend and follower to the AlchemyAPI Entity Extraction API. If a result is received that indicates a location has been found in the user account location, GCL parses the result and adds the location to a list of friend and follower locations. In the second step, GCL utilizes Dbpedia to extract location information. GCL removes punctuation from each user location of each friend and follower and then converts the full user location to a Dbpedia URL. GCL also splits the user locations by spaces and converts each token to a Dbpedia URL. Requests are then sent to each Dbpedia URL. Just as the process described in Section 3.2, GCL collects any geographical coordinates contained by the Dbpedia entity received from the request result. These coordinates are also added to the list of friend and follower locations.

After coordinates are extracted from all friend and follower user locations, GCL needs to analyze the list of coordinates to determine a reasonable prediction for the location of the original tweet. Existing approaches [1] [16] [17] have mainly used probabilistic models to determine which

[6] https://dev.twitter.com/rest/public

friends and followers might be in the same geographical region as the user. We employ a different technique for analysis of friends and followers location. GCL takes the entire list of friends and followers' locations from both steps (i.e., the AlchemyAPI step and the Dbpedia step) and clusters all locations using DBSCAN, a density-based clustering algorithm [4]. We chose to use the density-clustering package for Node.js[7] to implement the DBSCAN clustering step.

Our parameters to the DBSCAN algorithm are 0.5 for the cluster radius and 2 for the minimum number of points to form a cluster. Our points are stored as latitude/longitude pairs, which are in units of degrees. We chose 0.5 for the radius because a radius of 0.5 degrees is approximately the size of an average city, so DBSCAN will cluster tweets within cities. We chose 2 for the minimum number of points to form a cluster because some Twitter accounts have few friends or followers, yet it is still beneficial to obtain some result for clustering. For example, for a user with 4 friends, if 2 of the friends live in the same location, GCL is able to make a prediction that the user lives near the cluster of 2 friends. The DBSCAN algorithm clusters points that are densely packed together and considers points in low-density regions to be outliers. Because the DBSCAN algorithm is able to ignore outliers by not including them in a cluster, GCL is able to ignore single locations that are far away (e.g., many people have a friend who lives in another city) and focus on friend and follower locations that are clustered together geographically.

Once DBSCAN clusters the locations, GCL looks for the largest cluster. We choose to pick the largest cluster because this cluster represents where most of the user's friends and followers are located. GCL chooses the midpoint of the cluster as the estimated location of the tweet. The midpoint coordinates are added to the list of location predictions for the tweet.

[7] https://www.npmjs.com/package/density-clustering

3.4 Topic-based Geolocation

In addition to using a combination of content-based geolocation and relationship-based geolocation, we chose to use topic-based geolocation as a possible prediction source for tweet location. In previous work [3], we extracted the topic of tweets, or concepts that the tweet is discussing. We clustered the tweets by topic and discovered where the clusters were located geographically. In this way, we could understand how topics that people are tweeting about differ from place to place.

GCL takes advantage of the topic clustering algorithm we previously defined and is able to predict the location of some tweets based on their topic. In order to extract the topic of a tweet, GCL utilizes the AlchemyAPI Concept Tagging[8] and Keyword Extraction APIs[9]. Both APIs provide results that contain ranked topics pertaining to the tweet. Figure 2 shows the extracted topic for the example tweet, which is discussing traffic.

After topics are extracted from the tweet, GCL clusters the tweet along with existing tweets. To determine which topic cluster the tweet should be matched with, GCL needs to determine which topic cluster contains the same topics as the tweet. Tweets in the same topic cluster are discussing the same topic, whether it be an event, popular celebrity, or news subject. GCL calculates a similarity score between the tweet and each tweet in each topic cluster. The similarity score shows the similarity between the content of two tweets. It is calculated based on whether the two tweets contain the same hashtag, as well as the ranked topics of the tweet. A topic cluster's similarity score is shown in Formula 1. Given that t is the original tweet, t_m is the mth tweet in the topic cluster, and n is the number of tweets in the topic cluster, the topic cluster's similarity score is the average of the similarity scores between the tweet and all tweets in the topic cluster. The topic

[8] http://www.alchemyapi.com/api/concept-tagging
[9] http://www.alchemyapi.com/api/keyword-extraction

cluster with the highest similarity score determines which topic cluster matches the tweet. If no topic clusters exist with a similarity score above some threshold value, GCL does not match the tweet with any topic clusters, and the tweet is passed to the final step in the GCL pipeline.

$$similarity = \frac{\sum_{m=0}^{n} s(t,t_m)}{n} \quad \text{(Formula 1)}$$

If the tweet is matched to a topic cluster, GCL then determines whether that topic cluster is clustered in a geographic region or whether it is distributed evenly. If the topic cluster is clustered in one region, it can be predicted that the tweet's location is also within that geographic region. In order to determine whether the cluster appears more in one area, GCL uses an adapted TF-IDF algorithm. TF-IDF stands for Term Frequency-Inverse Document Frequency and is a statistic that determines how important, or meaningful, a word is to a document [19]. TF-IDF calculates the meaningfulness of a word by determining whether the word occurs many times across a large amount of text, or whether it is important to one piece of text. Our adapted TF-IDF algorithm determines whether a location occurs commonly throughout all topic clusters of tweets, indicating that the location simply has a higher population, or whether it occurs more within a specific topic cluster, indicating that that topic cluster is important to that location.

If the TF-IDF statistic indicates that the topic cluster appears in a geographical region, the coordinates of that region are placed into the list of possible predicted locations for the tweet. The tweet is then passed to the final step in the GCL pipeline.

3.5 Final Prediction

In the final step of the GCL pipeline, the list of estimated locations from the previous four steps is analyzed and a final predicted location is chosen. First, if any of the estimated locations are within .5° of each other, they are clustered together using the same DBSCAN algorithm as used by GCL in the friends and followers clustering step. If a cluster is found, this indicates that at least two

techniques produced predicted locations that were close together. GCL takes the average of the geographical coordinates in the cluster with the largest number of coordinates and considers the average location as the final predicted location. If no clusters are found, this indicates that either there is only one predicted location or the predicted locations are not within 0.5° of each other.

In the case where no clusters are found, GCL chooses the most likely estimated location. In order to discover which methods produced the most accurate predicted location most often, we conducted an experiment, described in Section 4. Following the results of this experiment, we created a method for choosing a final predicted location from the list of estimates. The predicted location is chosen from the highest probability method, shown in Table 1, that produced a candidate location.

4. Experiment

In this section, we describe our experimental setup and results.

4.1 Experimental Setup

Our pipeline is set up as a Node.js server. We ran GCL on a MacBook Pro with 16 GB RAM. Because Twitter API calls are restricted to the retrieval of 15 users' friends and followers list per 15 minutes, we decided to pre-fetch tweets along with each tweet's friends and followers list. Tweets were stored with their content, coordinates, userid (needed for retrieving friends and followers), and user location. We streamed geotagged tweets from the Twitter Streaming API[10] using twit[11], a node.js library for retrieving a Twitter stream. Friends and followers from each tweet

[10] https://dev.twitter.com/streaming/overview
[11] https://www.npmjs.com/package/twit

were collected from the Twitter REST API[12] also using twit. At the same time as collecting the tweets, we also ran our geotopical clustering system, as described in [3], to collect concepts within the stream for use in the topic-based geolocation step. The tweets and concepts were collected in October 2015.

We decided to use only geotagged tweets so that we could effectively analyze whether the location predicted by GCL was the user's actual location at the time of the tweet. This could produce a bias, because it is possible that the content of geotagged tweets contains more location information than non-geotagged tweets. However, using geotagged tweets is the only possible method for truly analyzing whether GCL can predict accurate locations. We believe after manually analyzing streams of non-geotagged tweets that any bias produced by using geotagged tweets is small.

After tweets were collected, we streamed the tweets as JSON objects through GCL. For each tweet, GCL analyzed whether both the final predicted location was correct and whether any of the estimated locations were correct. The results from this experiment are described in Section 4.2.

4.2 Experimental Results

In our evaluation, we present our results as two-fold: first, we analyze results we received where we consider all techniques equally for the final location. We check the user's actual location against all estimated locations for each tweet. If any of the estimated locations are correct, we count that tweet towards the count of correct estimations.

Second, we choose a final predicted location out of the list of estimated location coordinates. This is obviously a more real-world analysis than the first, because in a real system,

[12] https://dev.twitter.com/rest/public

one final location would usually need to be chosen, rather than several estimates of a user's location.

409 total tweets were run through GCL, including their user locations and friends and followers. We considered an "accurate" location prediction to be within 30 km, as this distance is fairly consistent with analyses conducted in existing research. GCL was able to gather the correct location within 30 km 51.83% of the time within one of the methods described in the pipeline. 212 tweets had a correct predicted location within at least one of its methods. This shows that for city-level accuracy, GCL is accurate about half of the time in predicting a location using any one of its methods.

For the second portion of the evaluation, we analyze the final predicted location from GCL. In order to calculate a final prediction out of the list of estimated location, we ran an experiment to evaluate which methods were most likely to produce an accurate predicted location. We ran 139 unique geotagged tweets through GCL and considered which techniques produced a result within 30 km of the tweet's actual location. The results are displayed in Table 1.

The rightmost column in Table 1 displays the percentage of the technique that was accurate out of the number of tweets that had an accurate result. This shows how much each particular method contributes to the overall accuracy. As shown, a tweet's friends and followers produce the correct location with the highest percentage. This is not surprising, because almost every Twitter account has at least one friend or follower, and many have quite a few. In contrast, not every user has an account location or mentions a location within their content. The related topic method is the next highest percentage of accuracy, and following that is the user account location and the tweet content. The results from this experiment were used to predict the final location.

We then ran the same 409 tweets from the first experiment through GCL and evaluated the final predicted location against the geotagged location. The overall prediction accuracy of the final prediction choice (comparing the final prediction by GCL to the geotagged location for each tweet) was 39.12% within 30 km, with 160 tweets having an accurate location prediction.

Using the friends and followers geolocation method contributed to an accurate prediction 65.09% of the time. User account location was the next most accurate method, contributing to the final prediction 60.38% of the time. Topic geolocation contributed to the correct location 18.87% of the time. This shows that topic geolocation can be an effective method, when combined with other existing methods, for predicting social media location.

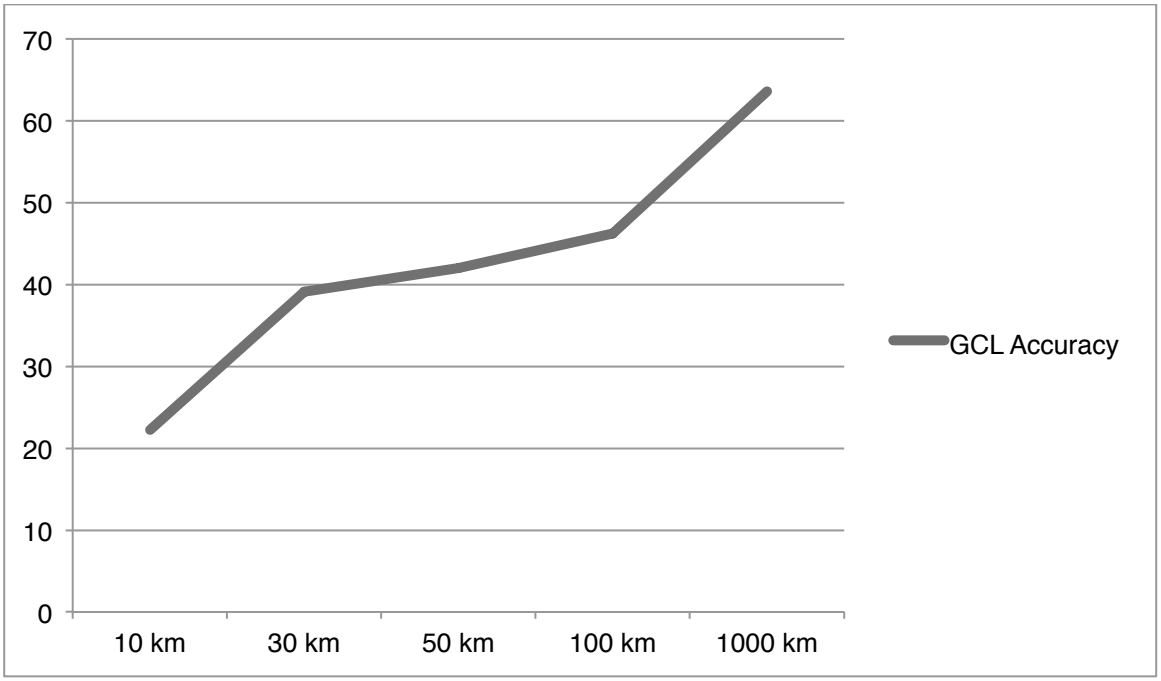

Figure 3

Figure 3 shows the accuracy of GCL across multiple distances. As displayed, 22.25% of total tweets analyzed were accurate within 10 kilometers. 42.05% of tweets were accurate within 50 kilometers. These percentages reflect the distance of the final predicted location with the actual geotagged tweet location.

Our results show that GCL is fairly accurate in predicting a location for a social media post. Although some existing approaches resulted in a higher accuracy, these methods were predicting the home location of a user. This is arguably an easier assumption to make, because users spend more time at their home location, so friends and followers plus the user account location can often produce a correct result for the home location. GCL attempts to predict the tweet's current location, however, which we argue is a more useful approach when performing social media analysis. If a user is on a trip away from their home location and is tweeting about their current location, the content of that tweet should be correlated with the current location, not the home location where the information may not be relevant.

5. Future Work and Conclusion

In this paper, we presented GCL, a new approach for geolocating posts from a social media stream. In the future, we plan to focus on improving our algorithm for geolocation in two main ways. First, we will improve the extraction of location-based terms out of the tweet's content. GCL is able to extract many locations that are well-known, such as cities, states, or large attractions, but it is not as effective at recognizing smaller venues such as restaurants. Second, we plan to improve the selection strategy for the final predicted location. Although the algorithm used by GCL in this experiment was fairly effective, there were a few times in which the final prediction location was inaccurate, even though one or more techniques produced an accurate location estimate, due to the selection strategy used by GCL to choose a final prediction.

GCL utilizes several different techniques for geolocation and combines those methods in an intelligent manner. It is able to geolocate 39.12% of tweets when run on our test set. GCL improves on existing geolocation approaches by using both friends and followers, as well as content and topical clustering as methods.

References

[1] Lars Backstrom, Eric Sun, and Cameron Marlow, "Find Me If You Can: Improving Geographical Prediction with Social and Spatial Proximity," in *19th International Conference on World Wide Web*, Raleigh, NC, 2010, pp. 61-70.

[2] Anuj Jaiswal, Wei Peng, and Tong Sun, "Predicting Time-sensitive User Locations from Social Media," in *Proceedings of 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, Niagara, Ontario, 2013, pp. 870-877.

[3] Elizabeth Williams, Jeff Gray, and Brandon Dixon, "Mobile Context Recommendations from Social Media through Geotopical Clustering," University of Alabama, SERG-2015-01.

[4] Martin Ester, Hans-Peter Kriegel, Jorg Sander, and Xiaowei Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Second International Conference on Knowledge Discovery and Data Mining*, Portland, OR, 1996, pp. 226-231.

[5] Kazufumi Watanabe, Masanao Ochi, and Rikio Onai, "Jasmine: a real-time local-event detection system based on geolocation information propagated to microblogs," in *20th ACM International Conference on Information and Knowledge Management*, Glasgow, Scotland, 2011, pp. 2541-2544.

[6] Yohei Ikawa, Miki Enoki, and Michiaki Tatsubori, "Location inference using microblog messages," in *21st International Conference on World Wide Web*, Lyon, France, 2012, pp. 687-690.

[7] Benjamin P. Wing and Jason Baldridge, "Simple supervised document geolocation with geodesic grids," in *49th Annual Meeting of the Association for Computational Linguistics*, Portland, OR, 2011, pp. 955-964.

[8] Timothy Baldwin et al., "A support platform for event detection using social intelligence," in *Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, Avignon, France, 2012, pp. 69-72.

[9] Eric Baucom, Azade Sanjari, Xiaozhong Liu, and Miao Chen, "Mirroring the real world in social media: twitter, geolocation, and sentiment analysis," in *International Workshop on Mining Unstructured Big Data Using Natural Language Processing*, San Francisco, CA, 2013, pp. 61-67.

[10] Bo Han, Paul Cook, and Timothy Baldwin, "Text-based twitter user geolocation prediction," *Journal of Artificial Intelligence Research*, vol. 49, no. 1, pp. 451-500, 2014.

[11] Solomon Kullback and Richard Leibler, "On Information and Sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79-86, 1951.

[12] Liangjie Hong, Amr Ahmed, Siva Gurumurthy, Alex Smola, and Kostas Tsioutsiouliklis, "Discovering Geographical Topics in the Twitter Stream," in *Proceedings of the 21st International Conference on World Wide Web*, Lyon, France, 2012, pp. 769-778.

[13] Stuart Gemen and Donald Gemen, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-6, no. 6, pp. 721-741, Nov. 1984.

[14] Quan Yuan, Gao Cong, Kaiqi Zhao, Zongyang Ma, and Aixin Sun, "Who, Where, When, and What: A Nonparametric Bayesian Approach to Context-aware Recommendation and Search for Twitter Users," *ACM Transactions on Information Systems*, vol. 33, no. 1, 2015.

[15] Zhiyuan Cheng, James Caverlee, and Kyumin Lee, "You Are Where You Tweet: A Content-Based Approach to Geo-locating Twitter Users," in *Proceedings of the 19th ACM Int'l Conference on Information and Knowledge Management*, Toronto, Ontario, 2010.

[16] Jeffrey McGee, James Caverlee, and Zhiyuan Cheng, "Location Prediction in Social Media Based on Tie Strength," in *Proceedings of the 22nd ACM Int'l Conference on Information and Knowledge Management (CIKM)*, San Francisco, CA, 2013, pp. 459-468.

[17] Rui Li, Shengjie Wang, Hongbo Deng, Rui Wang, and Kevin Chen-Chuan Chang, "Towards social user profiling: unified and discriminative influence model for inferring home locations," in *18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Beijing, China, 2012, pp. 1023-1031.

[18] Dominic Rout, Kalina Bontcheva, Daniel Preotiuc-Pietro, and Trevor Cohn, "Where's @wally?: a calssification approach to geolocating users based on their social ties," in *24th ACM Conference on Hypertext and Social Media*, Paris, France, 2013, pp. 11-20.

[19] Karen Sparck Jones, "A Statistical Interpretation of Term Specificity and Its Application in Retrieval," *Journal of Documentation*, vol. 28, pp. 11-21, 1972.

Technique	Number of Accurate Results	Percentage of Accurate Results
Friends and followers	43	57.75%
User account location - AlchemyAPI	19	26.76%
User account location – Dbpedia with one token	14	19.72%
User account location – Dbpedia with two tokens	13	18.31%
Content – AlchemyAPI	20	28.17%
Content – Dbpedia with one token	4	5.63%
Content – Dbpedia with two tokens	4	5.63%
Topic	22	30.99%

Table 1: Results from Technique Experiment

Figures